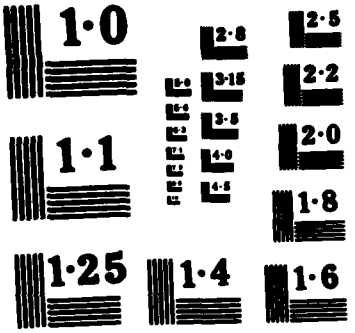
AD-A158 274 SPARSE ELIMINATION ON VECTOR MULTIPROCESSORS(U) 1/1
MICHIGAN UNIV ANN ARBOR SUPERCOMPUTER ALGORITHM
RESEARCH LAB D A CALAHAN 16 APR 85 AFOSR-TR-85-0542
UNCLASSIFIED AFOSR-84-0096 F/G 9/2 NL



(2)

Interim Technical Report

for Period 5-1-84 to 4-30-85

Grant AF-AFOSR-84-0096

SPARSE ELIMINATION ON VECTOR MULTIPROCESSORS

D. A. Calahan

Principal Investigator

Supercomputer Algorithm Research Laboratory

University of Michigan

Ann Arbor, MI

April 15, 1985

Unclassified
SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION: Unclassified	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY: ---	3. DISTRIBUTION/AVAILABILITY OF REPORT: Approved for public release; distribution unlimited
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE: N/A	
4. PERFORMING ORGANIZATION REPORT NUMBER(S)	5. MONITORING ORGANIZATION REPORT NUMBER(S): AFOSR-TR- 85-0542
6a. NAME OF PERFORMING ORGANIZATION: University of Michigan; 6b. OFFICE SYMBOL (If applicable):	7a. NAME OF MONITORING ORGANIZATION: AFOSR
6c. ADDRESS (City, State and ZIP Code): East Engineering Building, Ann Arbor, Michigan 48109	7b. ADDRESS (City, State and ZIP Code): Bldg. 410, Bolling AFB, D.C. 20332-6448
8a. NAME OF FUNDING/SPONSORING ORGANIZATION: AFOSR; 8b. OFFICE SYMBOL (If applicable): NM	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER: AFOSR-84-0096
8c. ADDRESS (City, State and ZIP Code): Bldg. 410, Bolling AFB, D.C. 20332-6448	10. SOURCE OF FUNDING NOS.

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT NO.
61102F	2304	A2	

11. TITLE (Include Security Classification): Sparse Elimination on Vector Multiprocessors

12. PERSONAL AUTHOR(S): D. A. Calahan

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Yr., Mo., Day)	15. PAGE COUNT
interim	FROM 1 Nov 84 TO 1 Apr 85	16 April 1985	6

16. SUPPLEMENTARY NOTATION

17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB. GR.	CRAY X-MP processors, LU decomposition, memory bank conflict
XXXXXXXXXXXXXX			

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

Studies of microtasking with up to 16 CRAY X-MP processors for LU decomposition of dense systems of equations have given rise to hybrid algorithms. One issue addressed has been the problem of memory bank conflicts, which increases with the number of processors. Conflict resistant algorithms have been developed. It is possible to assembly-code the X-MP so that accesses are pre-fetched into vector registers. Several reports have been prepared recently under this effort, and a paper entitled "Task Granularity Studies in a Many-Processor Cray X-MP" has been accepted for publication in Parallel Computing.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT	21. ABSTRACT SECURITY CLASSIFICATION	
UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT. ☒ DTIC USERS ☐	Unclassified	
22a. NAME OF RESPONSIBLE INDIVIDUAL: John P. Thomas, Capt, USAF	22b. TELEPHONE NUMBER (Include Area Code): (202)767-5026	22c. OFFICE SYMBOL: NM

DD FORM 1473, 83 APR — EDITION OF 1 JAN 73 IS OBSOLETE.

Unclassified
SECURITY CLASSIFICATION OF THIS PAGE

I. INTRODUCTION

In the development of concurrent algorithms for vector multiprocessors (VMP), a major distinction exists between small- and large-grain tasking. The latter is implemented at a high level of program logic (e.g., a physical space is divided into subspaces, each involving significant computation) from a high-level language, and will achieve a speedup nearly equal to p, the number of processors, for a variety of applications and for any VMP worthy of the name. Small-grain tasking is implemented at a low level to achieve similar speedups with more but smaller tasks. When this task size is at the vector operation level, it has been termed "microtasking" by this author [1]. Although early implementations of small-grain tasking were in assembly language on the Cray X-MP [1], Cray Research now feels that system software can be written with sufficiently small tasking overhead that similar performance can be achieved from Fortran. Even more important, a code such as the matrix multiply

```
      DO 1 I = 1, N2
      DO 1 J = 1, N1
    1 X(J) = X(J) + M(J,I)*Y(I)
```

could be <u>automatically</u> microtasked by a compiler by, for example, designating a task to be simply M(J,I)*Y(I) for all J. Such auto-tasking would not be possible in the large-grain, where global (likely problem-dependent) data dependencies are a consideration.

In summary, automatic small-grain tasking offers the ultimate in user convenience, with possibly modest loss in efficiency. It is clear that, where appropriate, small-grain algorithm studies on current VMP's should continue, although current implementations are from assembly language. These will challenge VMP architects to offer hardware to support such tasking (like the X-MP), rather than assume that only

large-grain tasking need be supported (like the Cray-2). We may suffer from the latter decision for many years.

II. PROGRESS REPORT

A. Introduction

The availability of instruction-level simulators for the X-MP and the Cray-2 is making possible the study of small-grain issues for many-processor VMP configurations. These include

(a) microtasking as indicated above, and

(b) designing conflict-resistant algorithms.

B. Hybrid-tasked algorithms

Studies of microtasking with up to 16 X-MP processors for LU decomposition of dense systems of equations have given rise to hybrid algorithms of the form of Figure 1. Here, a microtasked decoupling step is performed by all processors, to produce large-grain tasks which carry out the bulk of the computation. Originally simulated for a (hypothetical) architecture of Cray-1's connected to a common memory [1], a followup study has been made for a similar configuration of X-MP processors. The efficiency of the resultant algorithm is shown in Figure 2. An associated paper has been accepted for publication [2] and a similar study is being prepared for the Cray-2. It was pointed out by one reviewer that such examples may be considered as characterizing both the small- and large-grain attributes of an architecture in a single application. Of course, the computation model of Figure 1 has merit on its own right as one method of approaching problems not susceptible to large-grain tasking only.

C. Conflict-resistant algorithms

Both the X-MP and the Cray-2 have a design conflict between the need to conserve physical and wiring space (which translates into

speed) by minimizing the number of memory banks, and the need to isolate computations of processors connected to common memory and sharing memory banks. This results in an X-MP design which has a marginal conflict problem for 4 processors, and a serious degradation if 16 processors are used with a proportionately large number of banks. (The latter conclusion was a byproduct of the 16-processor algorithm studies above.) This result was considered of national significance, and an associated study was commissioned by Los Alamos National Laboratory [5].

This also gave rise to a diversion of our AFOSR grant resources into the new question of the effects of bank conflicts - expressed by access delays - on algorithm performance. (Early experience with the Cray-2 indicates this may be of overriding consideration.) An algorithm conflict sensitivity has been defined in [4] as

$$S = \frac{\text{total algorithm delay (clocks)}}{\text{average access delay (clocks)}}$$

which measures this effect. To reduce this ratio, it has been shown possible to assembly-code the X-MP so that accesses are pre-fetched into vector registers; ordinary delays in an access do not affect the later utilization of this data by the algorithm. These are examples of conflict-resistant algorithms. This concept has particular significance to library codes, which are often assembly-coded for speed; their incremental conflict sensitivity can often be reduced to zero. An associated report has been prepared [4].

III. COUPLING ACTIVITIES (WITH NATIONAL LABORATORIES)

A. Air Force Flight Dynamics Laboratory (FY 84-85)

Visiting scientist, studying vectorization of structural transient optimization codes, and (joint with AFOSR) VMP tasking of structural analysis algorithms.

B. Los Alamos National Laboratory (FY 84-85)

Consultant, studying conflict problems on a many-processor X-MP.

C. Lawrence Livermore National Laboratory - MFECC (FY 85)

Consultant, studying conflict problems on Cray-2.

D. NASA Ames Research Center (FY 84-85)

University research, developing high-performance libraries for the Cray-2.

E. Research Center for Advanced Computer Science (RIACS, NASA/Ames)

Consultant, assisting in the conversion of Computational Chemistry codes to the Cray-2, and the study of Cray-2 MP algorithms.

GRANT-SUPPORTED PUBLICATIONS

PREVIOUS YEARS

[1] Calahan, D.A., "Tasking Studies in Solving a Linear Algebra Problem on a CRAY-class Multiprocessor," Report SARL #5, Supercomputer Algorithm Research Laboratory, Dept. of Electrical Engineering and Computer Science, University of Michigan, December, 1983.

1984-85

[2] Calahan, D.A., "Task Granularity Studies on a Many-Processor Cray X-MP," accepted for publication in Parallel Computing.

[3] Calahan, D.A., "Influence of Task Granularity on Vector Multiprocessor Performance," Proc. 1985 Intl. Conf. on Parallel Processing, Bellaire, MI, August, 1985, pp. 278-84.

[4] Calahan, D.A., "Conflict Sensitivity of Algorithms," Report SARL #7, Supercomputer Algorithm Research Laboratory, Dept. of Electrical Engineering and Computer Science, University of Michigan, April 15, 1985.

OTHER PUBLICATIONS

[5] Calahan, D.A., "Memory Conflict Simulation of a Many-Processor CRAY Architecture. Part I: A CRAY X-MP Study," Report SARL #6, Supercomputer Algorithm Research Laboratory, Dept. of Electrical Engineering and Computer Science, University of Michigan, April 15, 1985.

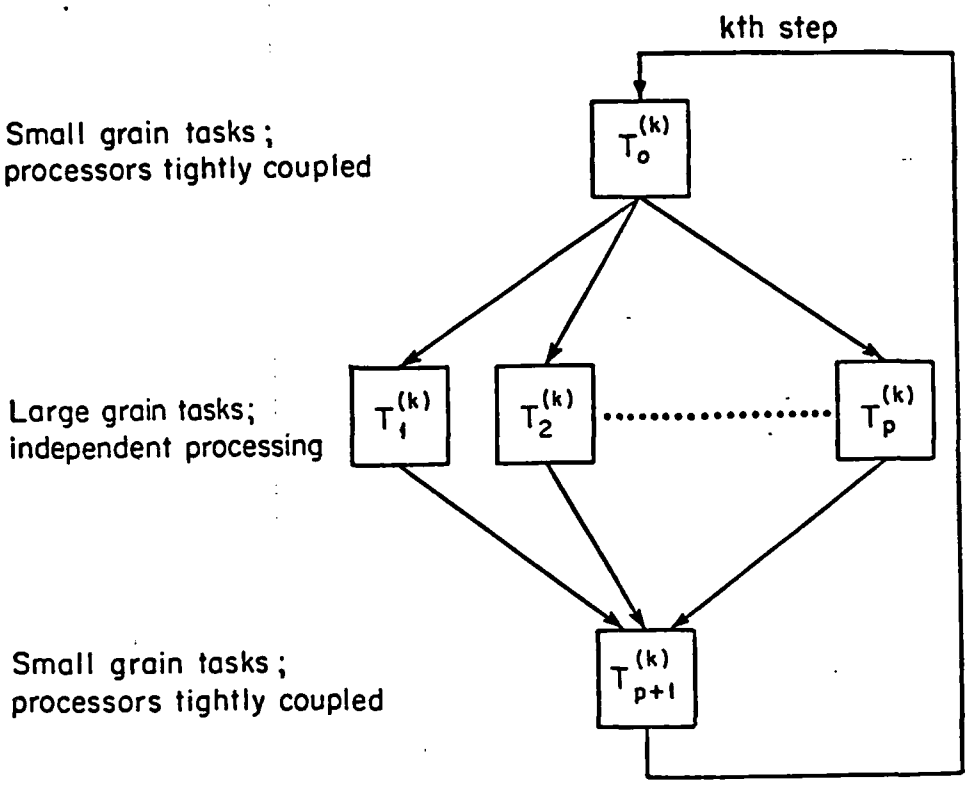


Figure 1. Hybrid granularity computational model

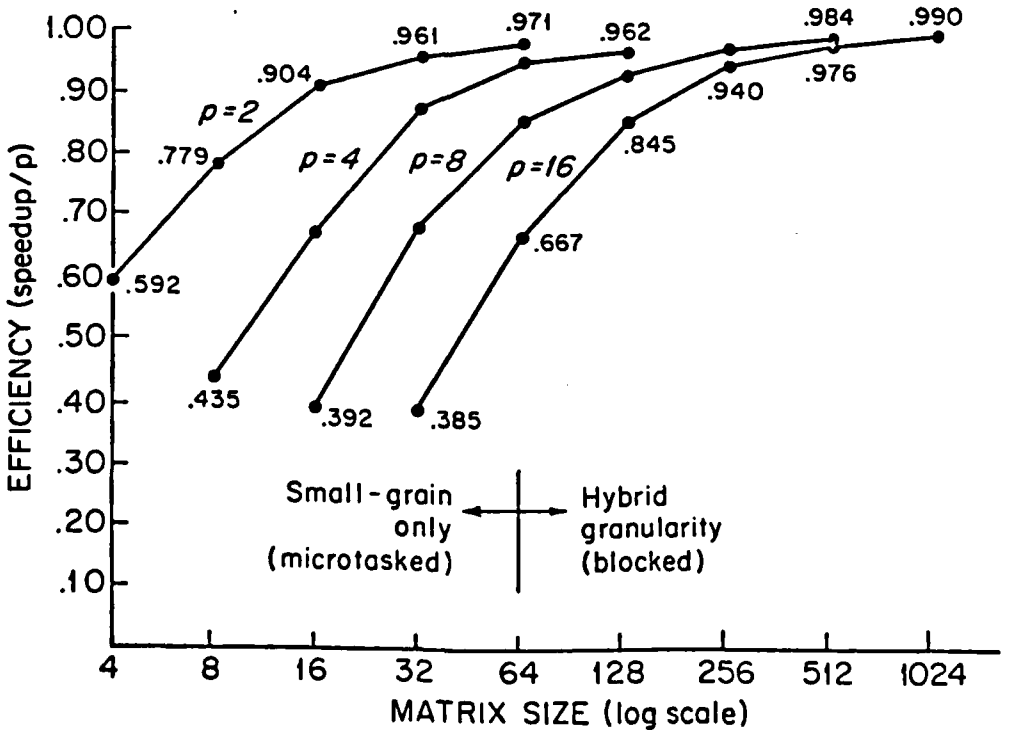


Figure 2. Performance of hybrid code

END

FILMED

9-85

DTIC